# 第四节 数据校验

## Bean Validation

### 核心 API

#### 元注解 - @javax.validation.Constraint

标注在目标校验注解上，来指定Bean Validation 校验器的实现

#### Bean Validation 校验器 - ConstraintValidator

用于实际的校验逻辑

#### 主要方法

**初始化方法 - #initialize**

通过注解方法获取相关的元信息

**校验方法 - #isValid**

通过对象传入，并且控制 ConstraintValidatorContext

**内建注解校验器实现**

- AssertFalse

通过 Message 文案了解当前 API 支持哪些内建注解，如 "javax.validation.constraints." 字符开头的

**自定义注解校验器实现**

## Bean Validation 校验器上下文 - ConstraintValidatorContext

- 获取错误文案（国际化）- getDefaultConstraintMessageTemplate()

## 文案解释器 - javax.validation.MessageInterpolator

# 校验分组

假设：@NotNull 在 User 类中的 name 上标注，但是存在以下情况：

- 如果在注册时，需要校验（group = REG)
- 如果在登录时，不需要校验 ( group = LOGIN)

```
@NotNull
private Strting name
```

# 引导 Bean Validation

## Bean Validation SPI - javax.validation.spi.ValidationProvider

## Bean Validation 配置 API - javax.validation.Configuration

# 作业

通过课堂上的简易版依赖注入和依赖查找，实现用户注册功能

- 通过 UserService 实现用户注册
- 注册用户需要校验
  - Id：必须大于 0 的整数

- 密码：6-32 位
- 电话号码：采用中国大陆方式（11 位校验）